Problem Solving by Computer: Project 2

April 3, 2020

1 Projectile Motion with Frictional Force

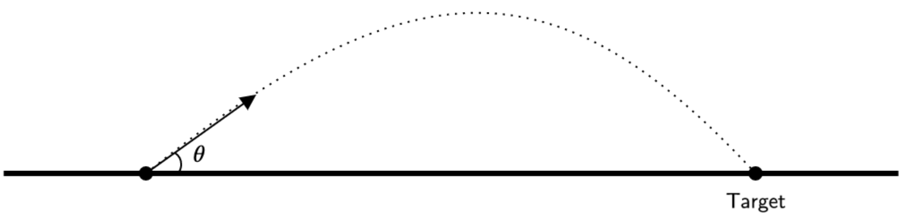
1.1 Description

Projectile motion is the form of motion an object takes after being projected from a point with an initial velocity and angle under the force of gravity. For example, we can observe or model a cannonball being shot from a cannon or a satellite orbiting the earth.

For the first part of this project, we had to calculate the maximum horizontal displacement of an object of weight 6kg being projected from a point with an initial velocity of 450ms^{-1}, along with the corresponding angle θ.

In the case where gravity is the only force acting on the object, the optimal angle is $\pi/4$, however in this problem we have another force acting on the object. There is a friction force $F = K|\vec{v}|^2$ also acting on the object, in the opposite direction of the velocity and proportional to the speed squared $|\vec{v}|^2$ with the constant K = 0.00002kgm^{-1}.

The figure below shows the setup of the model and the problem.



1.2 Methodology

For this problem, we need derive equations for the motion, starting from Newton's second law of motion $F = ma$. First, we will split the force acting on the object into horizontal and vertical directions. F_x will be our horizontal directional force and F_y will be our vertical directional force.

In the horizontal direction, there is only the horizontal component of the frictional force affecting the object, therefore $F_x = -K(\sqrt{V^2{}_x + V^2{}_y})V_x$, where V_x and V_y denote velocity in the horizontal and vertical direction respectively.

In the vertical direction, we have two forces acting on the object: the vertical component of the frictional force and gravity. Therefore, it can be stated $F_y = -K(\sqrt{V^2{}_x + V^2{}_y})V_y - mg$, where m is the mass of the object and g is the gravity of the Earth.

Now we can find the acceleration in both directions too by using Newton's second rule of motion. Acceleration in the horizontal direction can be denoted as $a_x = \frac{-K}{m}(\sqrt{V^2{}_x + V^2{}_y})V_x$. Similarly, in the vertical direction $a_y = \frac{-K}{m}(\sqrt{V^2{}_x + V^2{}_y})V_y - g$. We can see now that we have a pair of coupled ODEs to solve.

Using the MATLAB function ode45, we can use the equations we have derived as the arguments, along with initial conditions for each. We know that the initial velocities will be $450\cos(\theta)$ and $450\sin(\theta)$. Using the MATLAB function fminbnd, we find the θ which gives us the maximum horizontal displacement.

1.3 Algorithm

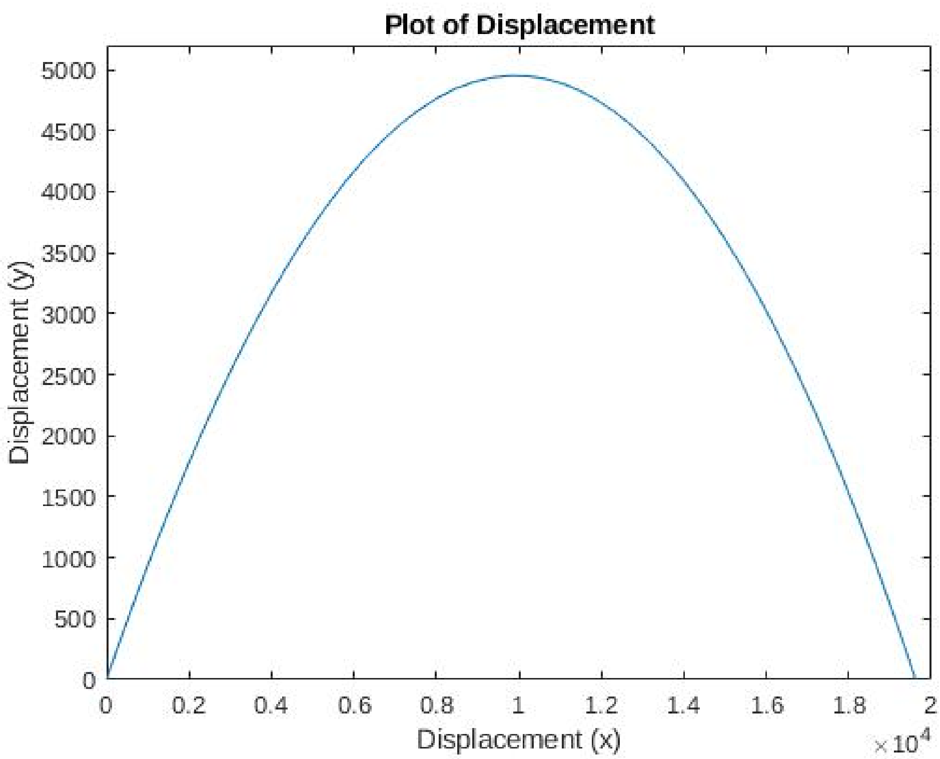
```
theta = fminbnd(@Dist, 0, pi/2) % Find the optimal theta
Dist(theta)

function d = Dist(theta)

% Find the max distance
maxDist = (450)^2 / 9.8;

options = odeset('events', @event_fun,'reltol', 1e-8);

%Coupled ODEs with their initial conditions

[t, q] = ode45(@(t,z) FrictionForce(t,z), [0, maxDist], [0, 0, 450*cos(theta),
    450*sin(theta)], options);

format long
d_max = q(end, 1);

%We want to find the maximum so we set this to minus
d = -d_max;

plot(q(:,1),q(:,2))
axis([0, 20000, 0, 5200])
end

%Event function to terminate when the y-value is equal to zero
```

```
function [value, isTerminal, direction] = event_fun(t,z)
value = z(2);
isTerminal = 1;
direction = -1;
end

%Function describing the coupled ODEs
function q = FrictionForce(t,z)
k = 0.00002;
g = 9.8;
m = 6;

q = [z(3); z(4); -k*((sqrt(z(3)^2 + z(4)^2)*z(3)))/m; -k*((sqrt(z(3)^2 +
    z(4)^2)*z(4)))/m - g];
end
```

1.4 Analysis

The optimal theta in radians for this projectile to reach the maximum horizontal displacement is 0.7784. Thus, the maximum displacement is 19617.76m.

The figure below describes the trajectory of the object in motion.



For the upper bound of the maximum displacement I used the maximum displacement when there is no frictional force acting on the object. This can be found using the formula $d = \frac{v^2}{g}$.

Since we are looking for the maximum and the MATLAB function fminbnd returns the minimum we must look for the negative, such that we can find the maximum. In addition, this function uses Golden-section search to find the best outcome. While in our case it is a sufficiently accurate method, we cannot overlook the fact that it is a approximation and there is a degree of error present.

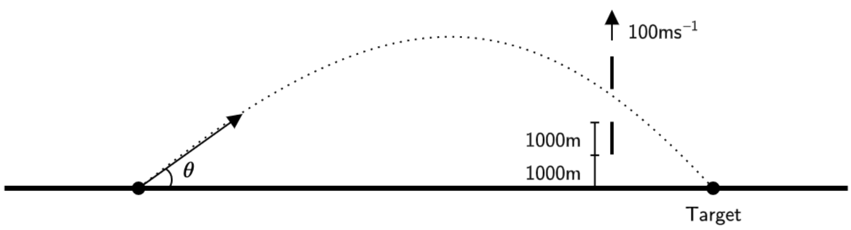
2 Projectile Motion with Interceptors

2.1 Description

This problem can be seen as a continuation of the previous one. However, instead of looking for the maximum horizontal displacement, we want to hit a target at 15,000m. Furthermore, at 12,000m, there are interceptors of length 1000m moving vertically at a rate of 20ms^{-1}. A new interceptor is launched every 20 seconds, therefore we need to time the projectile shot, such that the object does not hit the interceptor.

For this problem, the task is to find from which angles the object can hit the target at the specified distance and at which times can we shoot the object, such that it doesn't hit the interceptors.

The projectile motion and the interceptors can be seen in the figure below.



2.2 Methodology

The method for this problem is also derived from the previous one. First we find the number of angles which can hit 15000m. By plotting angle against distance from the first problem, we can see that each distance can be hit from only two angles. Using this information and the MATLAB function fzero, we can find the two angles which are able to reach the target.

Once we have found the angles, we know need to find the time at which they are in the path of the interceptors. This is simple to find using MATLAB functions as we just needed to stop computing once the object reached 12000m.

Now we have to solve the problem of the interceptors. Since, we know the time it takes for the object to travel to the path of the interceptors, we now need to find the time intervals at which we can shoot the objects.

2.3 Algorithm

```
format long

%Find the two thetas at 15000m
theta1 = fzero(@(theta) Dist(theta) - 15000, 0.4)
theta2 = fzero(@(theta) Dist(theta) - 15000, 1.2)

Dist2(theta1)
Dist2(theta2)

function d = Dist(theta)
maxDist = (450)^2 / 9.8;
options = odeset('events', @event_fun,'reltol', 1e-8);

[t, q] = ode45(@(t,z) FrictionForce(t,z), [0, maxDist], [0, 0, 450*cos(theta),
    450*sin(theta)], options);

d_max = q(end, 1);
d = d_max;

plot(q(:,1),q(:,2))
axis([0, 20000, 0, 5200])
end

function d = Dist2(theta)
maxDist = (450)^2 / 9.8;
options = odeset('events', @event_fun2,'reltol', 1e-8);

[t, q] = ode45(@(t,z) FrictionForce(t,z), [0, maxDist], [0, 0, 450*cos(theta),
    450*sin(theta)], options);

d_max = q(end, 1);
d = d_max;

plot(q(:,1),q(:,2))
axis([0, 20000, 0, 5200])
end

function [value, isTerminal, direction] = event_fun(t,z)
value = z(2);
isTerminal = 1;
direction = -1;
end
```

```
%Event function to terminate when the x value is 12000
function [value, isTerminal, direction] = event_fun2(t,z)
value = z(1) - 12000;
isTerminal = 1;
direction = 1;
end

function q = FrictionForce(t,z)
k = 0.00002;
g = 9.8;
m = 6;

q = [z(3); z(4); -k*((sqrt(z(3)^2 + z(4)^2)*z(3)))/m; -k*((sqrt(z(3)^2 +
    z(4)^2)*z(4)))/m - g];
end
```

2.4 Analysis

The aim of this problem was to find the angles and the times at which we could hit the targets. The two angles at which the object could hit the target at are 0.4254 and 1.1336 radians.

By finding the times the objects reach 12,000m (1.281 and 8.171 for each angle respectively) we can find the intervals of the times which the object can be shot easily as we know the rate at which the interceptors travel.

One of the interval relations is $[0, 1.281) \cup (11.281, 21.281) \cup (31.281, 41.281)$ and so on. The other is similarly $[0,8.171) \cup (18.171, 28.171) \cup (38.171, 48.171) \cup$... We can see that the intervals are separated by the time it takes for the interceptor to travel one whole length.